Assignment #3 – Implementing the Map ADT

Target Date: Friday, September 20 at 23:59

Submit your solutions to Moodle when completed.

DESCRIPTION

For this assignment, you will be creating two implementations of the Map ADT, using the approaches that we have discussed in class: Download the map[1].h and map[2].h files, which contain two versions of map.h, one for BST-based maps (Part 1), and the other for hash table-based maps (Part 2).

Part 1:

- Create a new C project, and provide the implementation of the map component (C strings to ints) using the binary search tree approach from the slides. Here, you are provided the interface information (i.e., function signatures) in the corresponding map[1].h file (this file should be renamed to map.h), along with the underlying data representation. You may use strcmp from the string.h library for determining the ordering of strings.
- Create another file called map_test.c, which creates one or more instances of the map you just defined, and thoroughly tests each of the functions.
- Submit your map.c and map_test.c files to "A3 – Part 1 - BSTMap" in Moodle.

Part 2:

- Create a second C project, copy over your map_test.c file from Part 1, and add the second map[2].h file (this file should be renamed to map.h) to the project. This version of map.h uses a hash table with separate chaining representation for map as discussed in the lesson. You should implement a reasonable (case sensitive) hash function for strings in your map.c file.
- Create a new map.c implementation file, which uses the underlying hash table representation given in map.h. It should start with INITIAL_BUCKETS buckets upon creation, and double the number of buckets and rehash all of the map entries whenever the actual load factor exceeds the MAX_LOAD_FACTOR.
- You should modify the map_test.c testing file from the previous part so that it also outputs the current load factor and bucket size standard deviation at relevant times – this way you have more "visibility" into the underlying data structure to help you make sure that things are working the way they should from an efficiency standpoint.
- Submit your new map.c file from this part to "A3 – Part 2 - HashTableMap" in Moodle.